Vector Field Design on Surfaces

Eugene Zhang, Konstantin Mischaikow and Greg Turk

Georgia Institute of Technology

Figure 1: This figure shows various vector fields created on surfaces using our vector field design system. The vector field shown at the right was used to guide texture synthesis shown in Figure 12 (right).

Abstract

Vector field design on surfaces is necessary for many graphics applications: example-based texture synthesis, non-photorealistic rendering, and fluid simulation. A vector field design system should allow a user to create a large variety of complex vector fields with relatively little effort. In this paper, we present a vector field design system for surfaces that allows the user to control the number of singularities in the vector field and their placement. Our system combines basis vector fields to make an initial vector field that meets the user's specifications.

The initial vector field often contains unwanted singularities. Such singularities cannot always be eliminated, due to the *Poincaré-Hopf index theorem*. To reduce the effect caused by these singularities, our system allows a user to move a singularity to a more favorable location or to cancel a pair of singularities. These operations provide topological guarantees for the vector field in that they only affect the user-specified singularities. Other editing operations are also provided so that the user may change the topological and geometric characteristics of the vector field.

We demonstrate our vector field design system for several applications: example-based texture synthesis, painterly rendering of images, and pencil sketch illustrations of smooth surfaces.

Keywords: Vector Field Design, Topology, Surfaces, Computational Geometry.

1 Introduction

Many graphics applications require an input vector field to achieve certain visual effects. For instance, example-based texture synthesis makes use of a vector field to define local texture orientation and scale. In non-photorealistic rendering, vector fields are used to guide the orientations of brush strokes and hatches. In fluid simulation, external force is a vector field which need not correspond to any physical phenomenon and can exist on synthetic 3D models. A vector field design system enables these applications to create many different visual effects by merely using different input vector fields. A vector field design system can also be used to test existing vector field visualization techniques [22, 23].

Vector field design refers to creating a continuous vector field on an input surface based on the user's specifications or application-dependent requirements. There are several challenges to the problem of vector field design on surfaces. First, the system should enable the user to create a wide variety of vector fields with a small amount of user input. Most existing vector field design system generate gradient-like vector fields, which limit their potential applications. Second, the user should be able to control the the number of singularities in the vector field and their placement. As pointed out in [15, 8], this is necessary for applications such as example-based texture synthesis and non-photorealistic rendering, in which unwanted singularities cause artifacts in the visual appearance. Third, the system should be as interactive as possible. Finally, to create a vector field design system for surfaces defined as meshes, we need to come up with a definition for vector field continuity on a mesh.

To achieve these goals, we propose a three-stage pipeline for vector field design. At the beginning, the user quickly creates a complex vector field without being concerned about its topology. Then, the system analyzes the vector field and provides visual feedback to the user. The user can then make controlled editing operations to the current vector field, such as moving a singularity or cancelling a pair of singularities. This process of iterative analysis and editing is repeated until the user is satisfied with the results.

2 Previous Work

Vector field *analysis* and *visualization* have been well studied, and a good survey is available in [4]. However, vector field *design* is far less explored.

There has been some prior work in creating a vector field on a surface. In all the instances that we know, such systems have been created in a quick manner to generate vector fields for a particular application, such as texture synthesis [15, 21, 24], fluid simula-

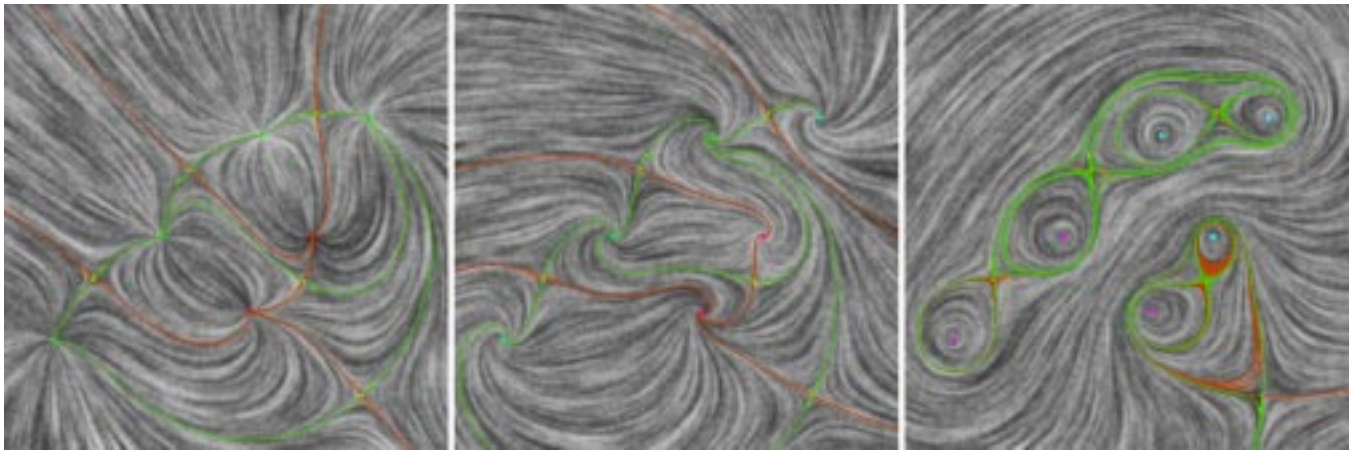

Figure 2: This figure shows three vector fields and their topological skeletons. The visualization is based on the image-based flow visualization technique by van Wijk [22]. The vector fields are curl-free(left), divergence-free(right), or neither (middle). The singularities are depicted using colored dots. Sources, sinks and saddles are colored in green, red and yellow. Repelling and attracting foci are colored in cyan and magenta. For saddles, their two principle directions are depicted using a cross.

tion [17], or for testing a vector field visualization technique [23]. There are three basic approaches for creating a surface vector field using these systems. In the first approach, a 3D vector field is created and projected onto the surface to obtain a tangential vector field [17, 23]. In the second approach, the user specifies desired vector values at a few locations on the surface and the system performs relaxation to obtain a global surface vector field [21, 24]. In the third approach, the user again specifies the vector fields at a few places on the surface. Then a global vector field is constructed by interpolating these locations using Gaussian radial basis functions over the surface [15]. These vector field design systems do not provide the user with controls over vector field topology, such as the number of singularities in the vector field and their locations. However, we borrow some of these ideas to create an initial vector field in the first of our three-stage design pipeline.

For planar domains, vector field design systems based on topological information have been demonstrated. van Wijk proposes a vector field design system to test and demonstrate his image-based flow visualization techniques [22]. In his design system, the user specifies desired singularity locations and types. The systems converts each specification into a globally-defined vector field and combines them into a global vector field using radial basis functions. However, vector fields created in this manner often have more singularities than those that the user has specified. Because this system does not provide a way of removing undesired singularities, it lacks control over vector field topology. Rockwood and Bunderwala [16] propose a technique in which the system uses geometric algebra to create a vector field based on user specified singularities locations and types (source, saddle, and etc). The user can interactively create a vector field by adding, editing and moving the singularities. This system also lacks the control over vector field topology since the vector field created this way may have unspecified singularities. On the other hand, Theisel [19] proposes a 2D vector field design system in which the users has the complete control over vector field topology. To do so, the user specifies the *topological skeleton* of the desired vector field and system creates piecewise-linear vector field to match it. This system requires the user to specify the desired vector field skeleton, which can be cumbersome for complex vector fields. Both topology-based design systems [16, 19] require a planar parameterization and cannot be generalized to work for curved surfaces in an obvious way.

All of these systems have certain traits that we wish to incorporate into our vector field design system. In fact, we borrow techniques from existing systems to serve our purpose at various stages. This will become clear in sections 4 and 6.

3 Background on Vector Fields

In this section, we review some basic facts about vector fields on surfaces. A vector field $\mathbf{V}$ for a manifold surface $\mathbf{S}$ is a smooth vector-valued function that associates to every point $\mathbf{p} \in \mathbf{S}$ a tangent vector $\mathbf{V}(\mathbf{p})$. A vector field defines a system of differential equations

$$\frac{d\mathbf{p}}{dt} = \mathbf{V}(\mathbf{p}). \tag{1}$$

For each point $\mathbf{p}^0 \in \mathbf{S}$, there exists a solution $\mathbf{p} : \mathbb{R} \to \mathbf{S}$ with the property that $\mathbf{p}(0) = \mathbf{p}^0$ [3, 9]. The set $\{\mathbf{p}(t) \mid t \in \mathbb{R}\}$ is called the *trajectory* through $\mathbf{p}^0$. Uniqueness of solutions to ordinary differential equations guarantees that the set of trajectories forms an equivalence relationship on $\mathbf{S}$. In particular, if $\mathbf{q}^0$ is an element of the trajectory of $\mathbf{p}^0$, then $\mathbf{p}^0$ is an element of the trajectory of $\mathbf{q}^0$.

Because of this relationship, the surface $\mathbf{S}$ can be decomposed into the set of all trajectories, though some trajectories are of particular significance.

Observe that if $\mathbf{V}(\mathbf{p}^0) = 0$, then the trajectory through $\mathbf{p}^0$ consists of a single point. In this case $\mathbf{p}^0$ is called a *singularity* (*fixed point*) of $\mathbf{V}$. Non-singular points are *regular*. Singular points can be further classified using the linearization of the vector field. To be more precise, a singularity $\mathbf{p}^0$ is *hyperbolic* if the real parts of the eigenvalues of $D\mathbf{V}(\mathbf{p}^0)$ are nonzero. Observe that if the two eigenvalues are complex numbers, then they must be conjugate, i.e., they have the same real part. A hyperbolic singularity is a *source* if both real parts are positive, a *sink* if both are negative, a *saddle* if one is positive and the other negative (this can only occur if both eigenvalues are real). Figure 2 shows three vector fields in a planar domain. The vector field is visualized using van Wijk's image-based technique [22]. Sources, sinks and saddles with real eigenvalues are colored in green, red and yellow dots. Sources and sinks with complex eigenvalues are colored in cyan and magenta.

Other trajectories of particular importance are *limit cycles* (*periodic orbits*) and *separatrices*. Limit cycles arise whenever the solution function $\mathbf{p}(t)$ is periodic. A separatrix is a trajectory for which the limit as $t \to \infty$ or $t \to -\infty$ of the solution function $\mathbf{p}(t)$ is a saddle.

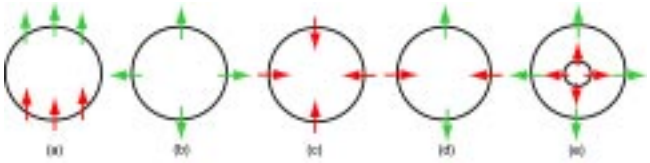

Figure 3: This figure shows five basic scenarios of isolating neighborhoods. Case (a), (b) and (e) are of particular interest since they are used when performing topological editing operations (section 5.4 and 5.5).

3.1 Analytic Descriptions

Two useful analytic characterizations of a vector field

$$\mathbf{V}(\mathbf{p}) = \begin{pmatrix} F(p_1,p_2) \\ G(p_1,p_2) \end{pmatrix}$$

are its curl and divergence:

$$curl(V) := \frac{\partial G}{\partial p_1} - \frac{\partial F}{\partial p_2}\,, \quad div(V) := \frac{\partial F}{\partial p_1} + \frac{\partial G}{\partial p_2}.$$

Basically, divergence refers to the amount the flow leaving and entering a small region near the measurement point, while curl measures the amount of flow that follows the region's boundary. Thus, the amount of curl or divergence exhibited by the vector field has an impact on the geometric structure of the trajectories. The two extreme cases are *gradient* vector fields in which case the curl is zero everywhere and *Hamiltonian* vector fields in which case the divergence is zero everywhere. It is important to observe that a general vector field, is neither curl-free nor divergence-free. (see Figure 2, middle).

3.2 Topological Descriptions

Our design problem requires that we be able to control the trajectories of a vector field both locally, even when the local analysis is degenerate, and globally. To do this requires introducing topological characterizations of vector fields.

A singularity $\mathbf{p}^0$ is *isolated* if there exists a compact neighborhood $U \subset \mathbf{S}$ of $\mathbf{p}^0$ with the property that $\mathbf{p}^0$ is the unique singularity in U. An isolated singularity $\mathbf{p}^0$ can be characterized by its *Poincaré index* which is defined as follows. Since $\mathbf{p}^0$ is isolated there exists a circle C which bounds a region in which $\mathbf{p}^0$ is the unique fixed point. The corresponding Gauss map α from C to the unit circle is given by $\alpha(\mathbf{p}) = \mathbf{V}(\mathbf{p})/|\mathbf{V}(\mathbf{p})|$ The number of times (keeping track of orientation) that the image $\alpha(C)$ covers the unit circle is always an integer and this number is the *Poincaré index* of the singularity $\mathbf{p}^0$. The index is 0 for any regular point, -1 for saddles, and 1 for sources or sinks (see [12] for further details).

For a continuous vector field $\mathbf{V}$ defined on a closed orientable manifold $\mathbf{S}$ with only isolated singularities, the *Poincaré-Hopf Index Theorem* tells us that the total indices of all the singularities of $\mathbf{V}$ equals the Euler characteristic of the surface $\mathbf{S}$. An immediate corollary of this is that given a particular vector field $\mathbf{V}$, if one wants to remove a singularity of positive or negative Poincaré index then one must simultaneously remove a singularity of the opposite sign. To perform these removals we borrow basic ideas from the Conley index theory (see [2, 13, 14] for further details and references) which is a topological generalization of Morse theory.

The following concept is the starting point for the Conley index theory. Given a region $N \subset \mathbf{S}$, let ∂N denote the boundary of N. Furthermore, given $\mathbf{p}^0 \in N$ let $\mathbf{p} : \mathbb{R} \to \mathbf{S}$ be the solution to (1) satisfying $\mathbf{p}(0) = \mathbf{p}^0$. The set of boundary points which leave or enter

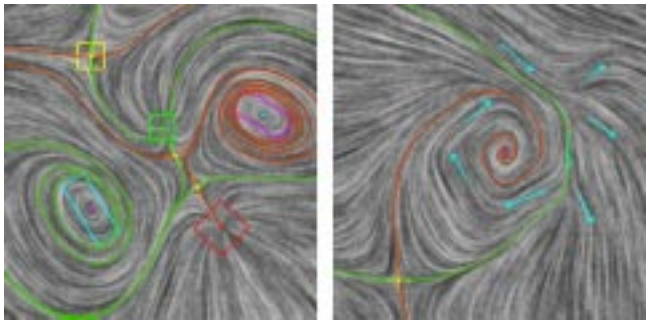

Figure 4: This figure shows two types of user specification for creating the initial vector field with *singular elements* (left) and *regular elements* (right).

N immediately can be characterized, respectively, by

$$N^- := \{\mathbf{p}^0 \in \partial N \mid \bigcup_{0<t<s} \mathbf{p}(t) \not\subset N,\ \forall s > 0\}$$
$$N^+ := \{\mathbf{p}^0 \in \partial N \mid \bigcup_{s<t<0} \mathbf{p}(t) \not\subset N,\ \forall s < 0\}.$$

A compact set N is an *isolating block* if $\partial N = N^- \cup N^+$. An isolation block is a region that has no points of internal tangency on its boundary, i.e., for each boundary point, there is either a forward or backward trajectory that leaves the region.

Given an isolating block N for a vector field $\mathbf{V}$, its Conley index is defined to be the relative homology [10] of N with respect to N^-, i.e. $CH_*(N) := H_*(N, N^-)$. For the purposes of this paper the computation of this index is fairly simple since our isolating block N will always take the form of a polygonal region and N^- will be a finite number of disjoint sets consisting of boundary edges of N. Idealized isolating blocks and their associated Conley indices are indicated in Figure 3. Observe that the Conley index is a finer invariant than the Poincaré index. In particular, the Conley index distinguishes between isolated sources and sinks. Figure 3 shows five scenarios of isolating neighborhood and the flow along their boundaries. Their Conely indices are as follows:

case (a) $CH_*(N) = 0$

case (b) $CH_*(N) = \begin{cases} \mathbb{Z} & \text{if } k = 2 \\ 0 & \text{otherwise} \end{cases}$

case (c) $CH_*(N) = \begin{cases} \mathbb{Z} & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$

case (d) $CH_*(N) = \begin{cases} \mathbb{Z} & \text{if } k = 1 \\ 0 & \text{otherwise} \end{cases}$

case (e) $CH_*(N) = 0$

Of particular interest are case (a), (b), and (e). We construct regions of these types for topological editing operations (section 5.4 and 5.5).

4 Design for Planar Domains

Our planar vector field design system consists of three stages:

1. Initialization: The user quickly creates a vector field with a set of specifications. At this stage, vector field topology is not a concern.
2. Analysis: The system performs both geometric and topological analysis of the current vector field and provides visual feedback to the user.
3. Editing: The user modifies the vector field through a set of pre-defined editing operations.

The user may perform many editing operations before accepting the result. The initialization and analysis stages are relatively straightforward and we describe them in section 4.1 and 4.2, respectively. The editing stage is at the core of our vector field design system and we will describe this in section 5.

4.1 Creating the Initial Vector Field

The first stage allows the user to easily create an initial vector field without being concerned about vector field topology. There have been two ways of creating such a field: relaxation [21, 24], and using basis vector fields [15, 22]. We adopt van Wijk's basis vector approach [22] because we are impressed by its intuitive nature and its simplicity. In this approach, every user's specification is used to create a basis vector field defined in the plane. Then the initial vector field is constructed as a weighted sum of these basis vector fields. There are two types of basis vector fields: *singular elements* and *regular elements*.

A singular element corresponds to a vector field that has a singularity of certain type at a desired location. For instance, if the user desires to have an isotropic source at location $\mathbf{p}_0 = (x_0, y_0)$ with strength $k > 0$, the system will create the following vector field for any point $\mathbf{p} = (x, y)$ in the plane as:

$$\mathbf{V}(\mathbf{p}) = e^{-d\|\mathbf{p}-\mathbf{p}_0\|^2} \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} x_\mathbf{p} - x_0 \\ y_\mathbf{p} - y_0 \end{pmatrix} \tag{2}$$

Here, d is a decay constant that is used to control the amount of influence of the basis vector field. Other isotropic singular elements include a sink, a saddle, a counter-clockwise center, and a clockwise center, whose matrices are the following:

$$\begin{pmatrix} -k & 0 \\ 0 & -k \end{pmatrix}, \quad \begin{pmatrix} k & 0 \\ 0 & -k \end{pmatrix}, \quad \begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & k \\ -k & 0 \end{pmatrix}. \tag{3}$$

The system allows the user to modify the scale, orientation and center location of each singular element as well as to remove an existing singular element. Modifications to singular elements will result in more complex matrices (details can be found in [22]).

A regular element refers to a vector field that has a particular nonzero vector value $\mathbf{V}_0$ at a desired location $\mathbf{p}_0$. Again, the system creates a basis vector field as follows:

$$\mathbf{V}(\mathbf{p}) = e^{-d\|\mathbf{p}-\mathbf{p}_0\|^2} \mathbf{V}_0 \tag{4}$$

The resulting vector field is interactively updated and displayed as the user continues to make adjustment to the set of regular and singular elements. Figure 4 shows two vector fields that were generated using singular elements (left) and regular elements (right). In practice, both types of specifications can be combined to create an initial vector field. Notice that summing the basis vector fields may cause unspecified singularities to appear. This will be handled by the topological editing operations described in section 5. The initial vector field is then sampled at the vertices and linearly interpolated inside the triangles.

4.2 Vector Field Representation and Analysis

We represent a vector field in the plane as a piecewise linear vector field. To be more specific, for a given planar triangular mesh, our system represents a vector field $\mathbf{V}$ by assigning vector values $\{W_1, W_2, ..., W_n\}$ at the mesh vertices $\{v_1, v_2, ..., v_n\}$. For a point $\mathbf{p}$ inside a triangle $T = \{v_{T_1}, v_{T_2}, v_{T_3}\}$ whose barycentric coordinates are $(\alpha_1, \alpha_2, \alpha_3)$, we have

$$\mathbf{V}(\mathbf{p}) = \sum_{j=1}^{3} \alpha_j W_{T_j} \tag{5}$$

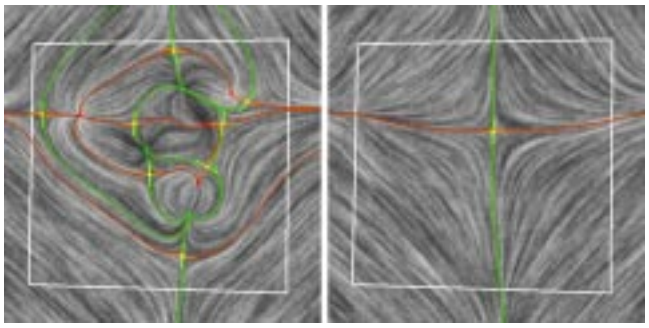

Figure 5: This figure shows the results of applying flow smoothing to a user specified region (in the white boundary). Notice the vector field defined outside the region is not changed.

or, under some local coordinate system of T,

$$V(\mathbf{p}) = \mathbf{M_T} \begin{pmatrix} x_\mathbf{p} \\ y_\mathbf{p} \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}, \text{ where } \mathbf{M_T} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{6}$$

This representation does not require an analytical formula and is compatible with many graphics applications that use vector fields. However, equations5 and 6 do not immediately generalize to a surface vector field representation (see section 6.2).

Once an initial vector field has been created, our system performs analysis on its analytical (geometric) and topological characteristics (section 3) and provides visual feedback to the user. For each triangle, the system computes the following information: divergence and curl, Poincaré index, the location of the singularity inside if any, and the incoming and outgoing directions if the triangle contains a saddle. Details of computing these quantities using the piecewise linear vector field representation can be found in [20]. We also compute the topological skeleton of the vector field. This is done by following the approach of Helman and Hesselink [6]. Starting from every saddle point, we follow the flow forward in its outgoing directions until the flow is stopped at a singularity or hits the boundary. To trace the trajectories away from a saddle we use a Runga-Kutta algorithm with adaptive stepsize control [1]. This gives us the two outgoing separatrices. Similarly, we obtain the two incoming separatrices by following the flow backward along the incoming directions of the saddle. Figure 2 shows the topological skeletons of the corresponding vector fields.

5 EDITING

The editing stage is at the heart of our vector field design system. The operations provided at this stage give a user detailed control over the vector field topology (the number of singularities and their positions) as well as the analytic characteristics such as divergence and curl at different places.

The editing operations that are useful will be application-dependent. When performing texture synthesis, for instance, the user may wish to remove unwanted singularities or to move them to less visible regions. Fluid simulation may require adjusting the amount of curl and divergence in the external force either locally or globally. We provide the following operations in our system: flow rotation and reflection, flow smoothing for a user-defined region, singularity pair cancellation, and singularity movement.

5.1 Flow Rotation

For any $\theta \in [0, 2\pi)$, we define an operator R_θ that acts on a vector field V to produce another vector field $R_\theta(\mathbf{V})$ as follows:

$$(R_\theta(\mathbf{V}))(\mathbf{p}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \mathbf{V}(\mathbf{p}) \tag{7}$$

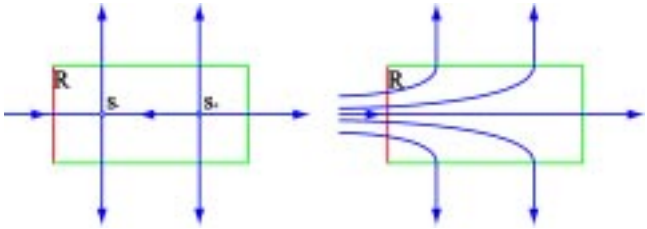

Figure 6: This figure shows the concept of singularity pair cancellation for a source $\mathbf{s}_+$ and a saddle $\mathbf{s}_-$. In the left portion, a region R is found to enclose both singularities and its boundary consists of two segments: inflow (red) and outflow (green). The vector field inside R is replaced with a flow that has no singularities (right).

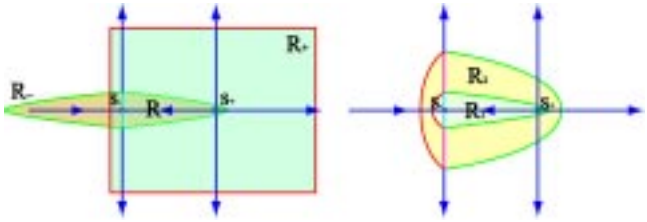

Figure 7: This figure shows our algorithm for construction the region R for singularity pair cancellation. In the left portion, we construct a region R_+ by following the flow forward from $\mathbf{s}_+$. Similarly, a region R_- is obtained by following the reverse flow from $\mathbf{s}_-$. When there is unique connecting orbits between $\mathbf{s}_+$ and $\mathbf{s}_-$, $R = R_+ \bigcup R_-$ is a region with trivial Conley Index. In the right portion, two valid regions R_1 and R_2 are obtained by using different sizes of R_-. R_2 is preferred since it is larger.

Simply, R_θ rotates the vector values by θ everywhere. It is easy to verify that

$$(curl(R_\theta(\mathbf{V})))^2 + (div(R_\theta(\mathbf{V})))^2 = (curl(\mathbf{V}))^2 + (div(\mathbf{V}))^2 \quad (8)$$

Topologically speaking, R_θ for any θ does not alter the number, the locations, or the Poincaré indices of the singularities in V. Any positive Poincaré indexed singularity can be converted into a source with appropriate θ. For a saddle, its incoming and outgoing directions are rotated by $\frac{\theta}{2}$. These properties makes the flow rotation operation essential for our system to perform topological editing operations such as singularity pair cancellation (section 5.4) and singularity movement (section 5.5), especially in regions of high curl.

5.2 Flow Reflection

Flow reflection is another vector field operator that does not alter the number or locations of the singularities in the original vector field. However, it negates the sign of their Poincaré indices. The reflection axis is arbitrary. If we use the y-axis as the reflection axis, the formula is:

$$(F(\mathbf{V}))(\mathbf{p}) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{V}(\mathbf{p}) \quad (9)$$

Just as flow rotation can convert a positive-indexed singularity into a source, flow reflection can turn any saddle into a source with appropriate choice of the reflection axis. This operation is important for the singularity movement operation (section 5.5) since we only need to develop algorithm for moving a source. Figure 11 shows the effect of this operation on for a vector field defined on a sphere. The concepts of flow rotation and flow reflection are not new, although we believe that our use of them for singularity movement and cancellation are novel. Theisel and Weinkauf [18] defined four types of vector field operations, including rotation and scaling (including reflection).

5.3 Flow Smoothing

The flow smoothing operation is simple yet efficient. Given a vector field $\mathbf{V}$ and a user-specified region R, it replaces the vector field $\mathbf{V}$ inside R with another vector field $\mathbf{V}'$ by solving the Laplace of the vector field inside R with $\mathbf{V}$ being fixed on the boundary. Let

$$\mathbf{V}'(\mathbf{p}) = \begin{pmatrix} F'(p_1, p_2) \\ G'(p_1, p_2) \end{pmatrix}$$

Then the new vector field $\mathbf{V}'$ is given by:

$$\begin{pmatrix} \varepsilon \bigtriangledown^2 F' = 0 \\ \varepsilon \bigtriangledown^2 G' = 0 \end{pmatrix} \quad (10)$$

in which ε is a spatial-varying function. In general, the choice of $\varepsilon = 1$ produces the desired result.

Note that the Laplace equation 10 is vector-valued, in contrast to the more familiar Laplace equation over a singular scalar value. This operation smooths the flow geometrically and will often reduce the number of singularities of $\mathbf{V}$ in R. In figure 5, a complex vector field with many singularities (left) is converted into a vector field with only one singularity (right) through smoothing. The user-specified region boundary is drawn in the white loop. Notice the vector field is not altered outside the user-specified region. We make use of the flow smoothing operation to perform topological editing operations such as singularity pair cancellation (section 5.4) and singularity movement (section 5.5).

5.4 Singularity Pair Cancellation

Our system provides the capability of eliminating or moving a singularity. As discussed in section 3, singularity elimination must be performed for a pair of singularities with opposite Poincaré indices. We therefore call this operation *singularity pair cancellation*. Our approach for this operation is as follows. First, we locate an isolating neighborhood R for the two singularities with trivial Conley Index. Second, we replace the flow inside R with one that does not have singularities. Figure 6 demonstrates the idea. This two-stage approach also applies when moving a singularity.

Let $\mathbf{s}_+$ and $\mathbf{s}_-$ be positive-indexed and negative-indexed singularities respectively. With proper flow rotation, $\mathbf{s}_+$ can always be converted into a source while $\mathbf{s}_-$ remains a saddle. From now on, we will assume that $\mathbf{s}_+$ is a source.

We also need the following definition. For a given vector field $\mathbf{V}$, let f denote the flow that is induced by $\mathbf{V}$. For a region S in the domain, we define its images under the forward and reverse flow as:

$$f(S) = \bigcup_{y \in S} \{f(y,t) | t \geq 0\}, \quad f^{-1}(S) = \bigcup_{y \in S} \{f(y,t) | t \leq 0\}. \quad (11)$$

To find the isolating neighborhood R for $\mathbf{s}_+$ and $\mathbf{s}_-$, we consider the following two regions $R_+ = f(M)$ and $R_- = f^{-1}(N)$, where M and N are some neighborhoods of $\mathbf{s}_+$ and $\mathbf{s}_-$ respectively. Furthermore, M and N are chosen such that they contain only one singularity in their interiors and no singularities on their boundaries. Then $R = R_+ \bigcap R_-$ is an isolation neighborhood. If there exists a unique separatrix going from $\mathbf{s}_+$ to $\mathbf{s}_-$ then the Conley index of $\mathbf{R}$ is trivial [14] (figure 7).

In practice, we choose M to be the triangle that contains $\mathbf{s}_+$. However, the choice of N is an delicate issue and its choice affects the shape of R_- and subsequently R . Due to the limited resolution of the underlying mesh, we prefer R to be as large as possible, so

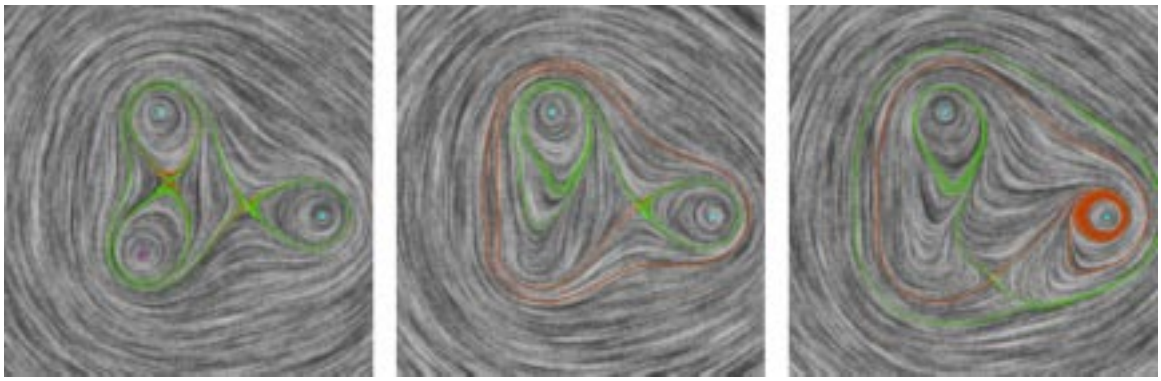

Figure 8: This figure shows the capability of our topological operations on a Hamiltonian vector field (left). First, a pair of center and saddle are cancelled (middle). Then, the remaining saddle is moved (right).

long as its Conley Index remains trivial. We perform a linear search on the length of the outgoing separatrices of $\mathbf{s}_-$ such that the covering triangles form N. Figure 7 shows the effect of following these separatrices to varying lengths.

To replace the flow inside R, we perform the flow smoothing operation (section 5.3) inside R while fixing the values on its boundary. As described before, this operation tends to simplify the vector field topology. We conjecture that flow smoothing for a region R with $\varepsilon > 0$, but sufficiently small, and with trivial Conley Index will always result in a singularity-free vector field inside R, and our numerical results support this conjecture.

Flow rotation is crucial for the success of the pair cancellation operation. If the original vector field has high curl around $\mathbf{s}_+$ as in the case of Hamiltonian system, then the connecting orbit between the singularities may not even exist. However, under appropriate rotations, the flow is converted into a vector field that has large divergence with a source at $\mathbf{s}_+$, and the connecting orbit becomes numerically stable. Once the pair cancellation operation is completed, a compensating rotation is performed.

5.5 Singularity Movement

Moving a singularity $\mathbf{s}$ to a new location allows the user to control the positions of the singularities in the vector field. Similar to singularity pair cancellation, this operation should only affect the intended singularity. Through flow reflection and flow rotation, we reduce the problem to moving a source. We use a two-stage algorithm that is similar to the pair cancellation. First, we construct a region R that encloses the connecting orbit for the current location $\mathbf{s_{old}}$ and the desired new location $\mathbf{s_{new}}$. By construction, R has the Conley Index of a source and does not contain any other singularities either in its interior or on its boundary (case (b) in figure 3). Second, we modify the vector field inside R such that the new flow has only one singularity at $\mathbf{s_{new}}$ (figure 9).

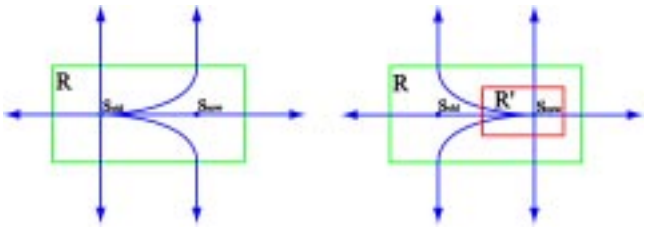

Figure 9: This figure shows the concept of moving a source from $\mathbf{s_{old}}$ to $\mathbf{s_{new}}$. A region R is found to enclose both $\mathbf{s_{old}}$ to $\mathbf{s_{new}}$, and R has the Conley index of a source. Then a small region R' is found to enclose $\mathbf{s_{new}}$ and vector values are assigned to the boundary of R' so that it forces a source at $\mathbf{s_{new}}$. For $R - R'$, flow smoothing operation obtains a flow without any singularity.

Let $R = R_+ \bigcap R_-$ where $R_+ = f(M)$ and $R_- = f^{-1}(N)$. Here, M is a small neighborhood of $\mathbf{s_{old}}$ and N is a neighborhood of $\mathbf{s_{new}}$. To ensure $\mathbf{s_{new}}$ is in the interior of R, we locate another point $\mathbf{s}'$ on the forward trajectory from $\mathbf{s_{new}}$ under $\mathbf{V}$. Then we consider the trajectory J of $\mathbf{s}'$ under the flow $R_{\frac{\pi}{2}}(\mathbf{V})$. J serves the same purpose as the separatrices for pair cancellation. We let N be the largest segment on J that makes R an isolating neighborhood with the Conley Index of a source. This ensures that R is a wide region.

We then consider the triangle T that contains $\mathbf{s_{new}}$ and assign vector values to force a source inside. The region $R' = R - \{T\}$ has two boundaries. The flow enters R' from the inner boundary and leaves at the outer boundary. R' therefore has trivial Conley Index (see Figure 3(e)) and flow smoothing inside R' produces a vector field without singularities.

We use rotation and reflection to make singularity movement applicable to a singularity of any index. If $\mathbf{s_{old}}$ is a saddle, we use reflection to turn it into a positive index singularity. If the vector field has high curl around $\mathbf{s_{old}}$, then we rotate the vector field so that the flow is converted to a vector field that has large divergence. This simplifies the location of a connecting orbit between $\mathbf{s_{old}}$ and $\mathbf{s_{new}}$.

Figure 8 shows the results of applying two successive topological editing operations to a Hamiltonian vector fields (left). First, a pair of center and saddle are cancelled (middle). Next, the remaining saddle is moved (right).

6 Design for Curved Surfaces

Now that we have presented our design techniques for planar domains, we describe how to extend these methods to surfaces. As in the case of planar domains, our system for surfaces consists of three stages: initialization, analysis, and editing. However, there are difficulties we must overcome. First, the tangent planes of mesh surfaces are discontinuous at the vertices and the edges, where the concept of vector field continuity breaks down. Second, a curved surface in general does not have a global parameterization. Yet, we need surface parameterization to correlate vectors defined at different locations. In the following sections, we propose a definition of *vector field consistency* for mesh surfaces. Furthermore, we make use of the concept of *parallel transport* to set up the correlation between the local frames of different tangent planes. These changes allow us to adapt our planar vector field design system to surfaces.

6.1 Vector Field Consistency on Mesh Surfaces

For planar domains, the concept of vector field continuity is well-defined since the tangent planes from different locations coincide.

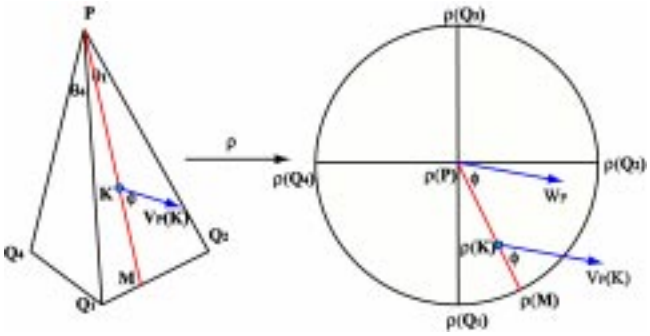

Figure 10: This figure shows the idea of parallel transporting vector W_i from vertex $v_i = \mathbf{P}$ to a point $\mathbf{K}$ inside $\mathbf{P}$'s 1-ring neighborhood, U. First, we build a local parameterization ρ for U. Then, we parallel transport the vector $W_\mathbf{P}$ to $\mathbf{K}$ along ray $\overrightarrow{\rho(\mathbf{P})\rho(\mathbf{K})}$. This construction guarantees the vector field consistency on surfaces.

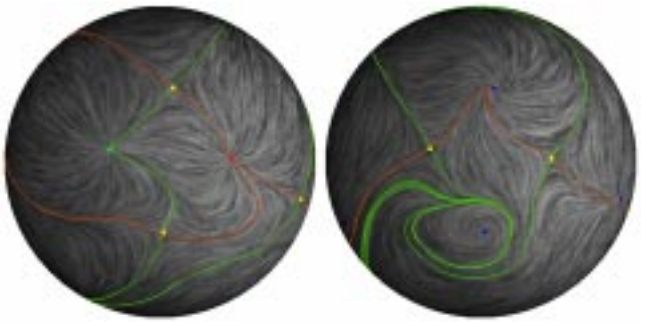

Figure 11: This figures shows the effect of performing flow reflection to a vector field (left) on the sphere. Notice the locations of the singularities do not change. Rather, the sign of their Poincaré indices are negated.

Furthermore, tangent vectors at different locations can be identified or compared using the Cartesian coordinate system. On a mesh representing a curved surface, neither fact holds. On the other hand, for any continuous surface vector field and a point $\mathbf{p}$ on the surface, we expect one of the two following situations to occur around an arbitrarily small neighborhood of $\mathbf{p}$. First, $\mathbf{p}$ is a singularity with respect to the vector field. Notice zero vectors at different locations can always be compared. Second, the flow induces a set of non-intersecting and space-filling trajectories, exactly one of which contains $\mathbf{p}$. Notice that as spatial curves, it makes sense to discuss the continuity and intersections of trajectories. Based on these observations, we propose the following definition.

Assume that vector field $\mathbf{V}$ is defined on a mesh surface $\mathbf{M}$. $\mathbf{V}$ is *consistent* at a point $\mathbf{p} \in \mathbf{M}$ if one of the following situations is true:

(a) For any *path* $\gamma([0,1)) \rightarrow \mathbf{M}$ where $\mathbf{V}(\gamma(t))$ is well defined and $\lim_{t\rightarrow 1}\mathbf{V}(\gamma(t))$ exists, $\lim_{t\rightarrow 1}\mathbf{V}(\gamma(t)) = 0$. In this case, $\mathbf{p}$ is called *singular*.

(b) There exists a neighborhood U of $\mathbf{p}$ and a homeomorphism $h : U \rightarrow \mathbb{R}^2$ which carries each piece of a trajectory lying in U onto a straight line in $\mathbb{R}^2$ parallel to the x-axis. In this case, p is called *regular*.

In other words, a consistent vector field on a mesh surface should exhibit the same local behavior as those defined on a plane.

6.2 Vector Field Representation and Analysis

As in the case of planar domains, we represent a vector field V by assigning vector values $\{W_1, W_2, ..., W_n\}$ at the mesh vertices $\{v_1, v_2, ..., v_n\}$. However, we cannot simply perform linear interpolation since the W_i's are in general not co-planar. Furthermore, without surface parameterization, tangent vectors that are defined at different locations can not be summed. To over these problems, we first define a parameterization for the 1-ring neighborhood of every vertex v_i. This parameterization allows us to parallel transport W_i to any point $\mathbf{p}$ inside v_i's 1-ring neighborhood. Let μ_i be such transport function (which we will soon describe). Then, for a point $\mathbf{p}$ inside a triangle $T = \{v_{T_1}, v_{T_2}, v_{T_3}\}$ whose barycentric coordinates are $(\alpha_1, \alpha_2, \alpha_3)$, we can rewrite equation 5 as follows:

$$V(\mathbf{p}) = \sum_{j=1}^{3} \alpha_j \mu_{T_j}(W_{T_j}, \mathbf{p}) \tag{12}$$

This results in a consistent vector field over the surface based on the values of W_i. We will now describe the parameterization and the transfer function in detail. In figure 10 (left), $\mathbf{P} = v_i$ is a vertex with the tangent plane $TM_\mathbf{P}$ (right). Its 1-ring neighborhood R consists of the triangles $\triangle\mathbf{PQ}_1\mathbf{Q}_2$, $\triangle\mathbf{PQ}_2\mathbf{Q}_3$, ..., and $\triangle\mathbf{PQ}_n\mathbf{Q}_1$ ($n = 4$). Let $\theta_i = \angle\mathbf{Q}_i\mathbf{PQ}_{i+1}$, $\theta = \sum_{i=1\rightarrow n}\theta_i$ and $r = \frac{\theta}{2\pi}$. Notice that $r = 1$ for vertices with zero Gaussian curvature. In the figure's right portion, let D be the unit disc in $TM_\mathbf{P}$ and let ρ be the following homeomorphism from R to D. First, ρ maps any ray from $\overrightarrow{\mathbf{PQ}}$ where $\mathbf{Q}$ is on D's boundary to a radius of $\overrightarrow{\rho(\mathbf{P})\rho(\mathbf{Q})}$. Along each ray, ρ is a linear map. Second, let $\phi_i = \angle\rho(\mathbf{Q}_i)\rho(\mathbf{P})\rho(\mathbf{Q}_{i+1})$. Then $\theta_i = r\phi_i$. For a ray $\overrightarrow{\mathbf{PM}}$ inside triangle $\triangle\mathbf{Q}_i\mathbf{PQ}_{i+1}$, we have $\angle\mathbf{Q}_i\mathbf{PM} = r\angle\rho(\mathbf{Q}_i)\rho(\mathbf{P})\rho(\mathbf{M})$. Note that this construction is similar to the "geodesic polar map" that is used by Welch and Witkin [25] for free-form shape design, with a minor difference: in their setting the parameterization domain is a polygon, not the unit disc as in our case.

To transfer W_i to a point $\mathbf{K}$ inside triangle $\mathbf{Q}_i\mathbf{PQ}_{i+1}$, we first locate the ray $\overrightarrow{\mathbf{PM}}$ that contains $\mathbf{K}$. Let ϕ be the counter-clockwise angle between W_i and the ray $\overrightarrow{\rho(\mathbf{P})\rho(\mathbf{M})}$, then we define $\mu_i(W_i, \mathbf{p})$ as the vector at $\mathbf{K}$ such that the angle between $\mu_i(W_i, \mathbf{p})$ and $\overrightarrow{\mathbf{PM}}$ equals ϕ. Furthermore, $||\mu_i(W_i, \mathbf{p})|| = ||W_i||$.

As a parameterization, ρ does not distinguish between points inside a triangle or on an edge. Therefore, vector field continuity is automatically guaranteed. Furthermore, the continuity of ρ ensures the continuity of the resulting vector field at the vertices. When $r = 1$ everywhere, this becomes the piecewise linear representation that we have used for planar vector field. This representation still permits analytical and topological analysis for surface vector fields. Figure 1 and 11 show the topological skeletons of various surface vector fields that are represented in this manner.

6.3 Creating the Initial Vector Field

To create an initial vector field on a surface, the user specifies singular and regular elements, which are then converted into basis vector fields and summed. This requires us to be able to globally parameterize the surface and to synchronize the local frames at every point. Neither is possible for a general manifold surface. However, since the focus of each user-specified element is the nearby region of the desired location, we create a parameterization that is of high quality near the element but that may have discontinuities at distant points.

We again face the task of parallel transport of vectors to far away regions, and we need a global polar map (ρ, θ) for this. For a given seed point on the surface (the location of a user-defined element), we compute the distance and angle from the seed to all other points on the surface. To be concise, let γ be the segment of a geodesic inside the triangle that contains the seed. This segment must be a straight line segment. Its angular parameterization in the tangent plane of the seed point will be the angle for any point on the geodesic. On a smoothed surface, there is a shortest geodesic

Figure 12: This figure shows the results of applying our surface vector field design system to texture synthesis. Two vector fields are used as input. In the left image, a vector field was created by placing a sink at the center of the bunny's tail. This vector field has several unwanted singularities on the bunny's front and paws, and these singularities lead to breakup of the texture pattern. The vector field used for the middle and right images consists of two singular elements: a source and a sink, both on the visible side of the bunny in the right portion. This vector field is then rotated by 60 degrees to create spirals. The resulting vector field has no singularities on the bunny's front side. The synthesis has no visual artifacts like the ones caused by the first vector field.

connecting any two points. This is in general untrue for a mesh surface. However, for places where there are no connecting geodesic to the seed point, we can assign any values. Since the vectors are weak in those places, the user can easily regularize these regions by adding elements nearby.

The distance component ρ of the polar map can be computed using the fast marching method proposed by Kimmel and Sethian ([11]). For the points that are near the seed, we obtain the angular component θ by projecting them onto the seed's tangent plane. We then propagate θ to points that are far away via minimal geodesics. For most parts of the surface, θ is continuous. To compute the vector value at a vertex with respect to an element, we can use equation 2 and 4 to produce the vector in the seed's tangent plane before parallel transporting it into the desired vertex location. Alternatively, we record the parallel transport of the local frames at the seeds to every vertex. This enables us to create basis vector fields by evaluating equations such as 2 and 4 directly (a source and a regular element, respectively) .

Let us stress that this is not the only way to create basis vector field. In van Wijk's visualization tool [23], an element is translated into a 3D vector field before being projected onto the surface. Constrained optimization is another way to produce an initial vector field with desired behaviors.

6.4 Editing

While the main concepts for editing operations on a surface remain the same as those on planar domains, some changes need to made to reflect the difference in vector field representation and the complexity of the surface geometry such as curvature and higher genus.

To perform global rotation on a surface, we simply rotate the vector field at each vertex. Notice that flow rotation for surface vector fields does not require any global parameterization. On the other hand, flow reflection requires that the local frames and reflection axes at each vertex are correlated. We again make use of a global polar map to parallel transport this information. Global reflection swaps the sign of the Poincaré indices for singularities without changing their locations.

We also adapt our flow smoothing operations to surface vector fields. For a given region R, we compute the polar map whose seed is the center triangle $\mathbf{T}$ of R. This allows us to obtain the parallel transport function $\mu_{\mathbf{T}}$. Subsequently, we solve equation 10 in which p_1 and p_2 are the polar coordinates of every vertex and F' and G' are expressed using $\mu_{\mathbf{T}}(W_i, v_i)$. Once finished, the resulting vectors are transported back onto each vertex via $\mu_{\mathbf{T}}^{-1}$. We make use of the flow smoothing operation for singularity pair cancellation and movement on surfaces.

7 Results

We have applied our vector field design system to several applications: painterly rendering of images, pencil sketch illustration of smooth surfaces, and example-based texture synthesis.

There have been numerous published approaches to painterly rendering, and to review them is beyond our scope. In this work, we use the approach by Hertzmann [7] with the following modification: instead of using the image gradient field to guide the brush stroke orientations, the user creates a synthetic vector field with our vector field design system. To make this task fast and effective, we incorporate the painterly rendering algorithm into our vector field design system. In addition to viewing the vector field, the user can also switch to the painterly rendering that uses the current vector field. The results are interactively displayed as the user makes changes to the vector field. Figure 14 shows some painterly rendering results. The final high-quality painterly images in this figure are created off-line using the algorithm of Hays and Essa [5].

We have modified van Wijk's image based flow visualization technique [23] to create non-photorealistic illustration of surfaces. Figure 13 shows the results of applying this techniques on the feline, Venus, and the horse.

Figure 12 shows the results of applying our vector field design system to texture synthesis. The texture synthesis method is based on [21, 24]. Two vector fields are used in this figure. The vector field used for producing the left image contains several unwanted singularities on the front side of the bunny, and these unwanted singularities lead to breakup of the texture pattern. The vector field used for the middle and right images contains no singularities in the same region. The texture result has no visual artifacts liked the ones caused by the first vector field. Also, the spiralling in the second vector field near the singularities on the side of the bunny is evident in the texture in the right image.

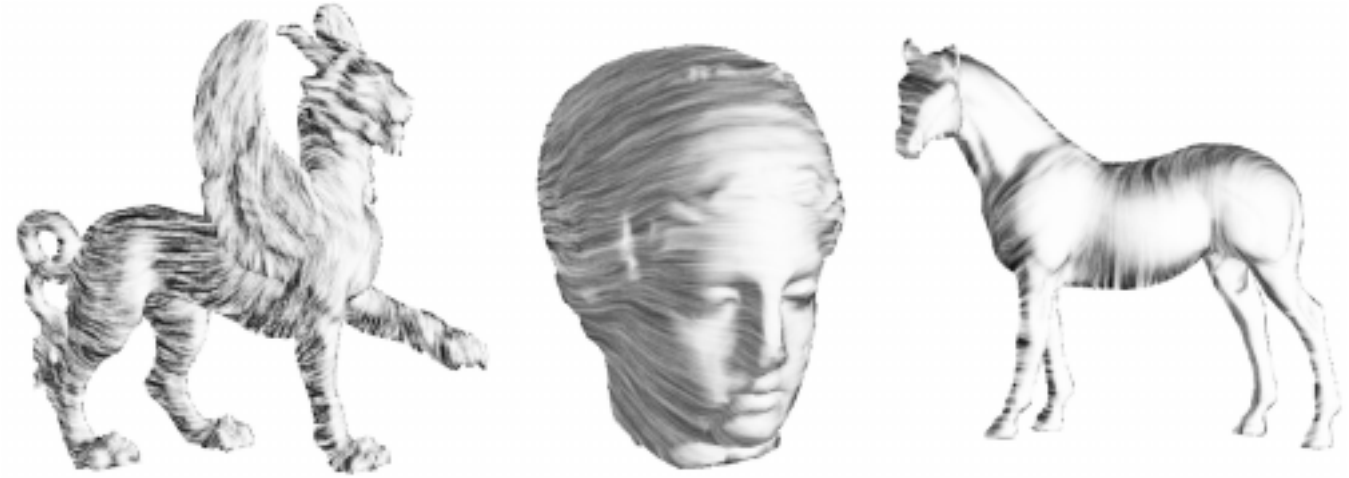

Figure 13: This figure shows the results of applying our vector field design system to non-photorealistic illustrations of surfaces. The pencil style illustration is based on van Wijk's image-based flow visualization technique [23].

8 Conclusion and Future Work

In this paper we have described an interactive vector field design system for surfaces. The system has the following attributes:

- It allows the user to create general vector fields, not just a subclass of vector fields such as divergence-free or curl-free vector fields.
- It allows the user to generate complex vector fields with a relatively small amount of user input.
- The user has control over vector field topology, including the number of the singularities and their placement.
- It works for both planar domains and curved surfaces.

To the best of our knowledge, this is the first system that possesses all of these characteristics. To give the user control of vector field topology, we borrow results from the Conley index theory to perform topological operations such as singularity pair cancellation and movement. The Conley index theory is well known in the dynamical systems community. However, as far as we know, this is the first time it has been used for a computer graphics application. We have proposed a new vector field representation on surfaces that is based on the geodesic polar map and parallel transport (section 6.2). This representation guarantees vector field consistency. We have also introduced a new method of creating basis vector fields on surfaces (section 6.3). We have demonstrated the utility of our design approach with three specific applications: painterly rendering, texture synthesis, and pencil sketch illustrations.

There are a number of issues that we wish to improve upon in our current system. First, our system uses the same decay constant d for all design elements for creating basis vector field (equations 2 and 4). It may be desirable to let the user control this. Second, our algorithm for building an isolating neighborhood sometimes produces regions that are larger than necessary. This means that the behavior of the flow may be changed at places that are far away from the user-specified singularities. We plan to investigate ways of restricting the size of such regions. Third, our system requires the user to specify the pair of singularities for cancellation. It would be nice to provide the functionality "singularity elimination", in which the user specifies one singularity to be removed and allow the system to determine another singularity for pair cancellation.

There are many interesting issues for future research. First, we would like to extend our surface algorithm to handle manifolds with boundaries. Second, we would like to identify and implement more editing operations, such as removing limit cycles. Conley index theory is general enough to treat limit cycles, and we think it will be straightforward to extend our techniques to moving and cancelling them. Finally, we would like to use our techniques for other applications such as fluid simulation and hair style design.

References

[1] J.R. Cash, and A.H. Karp, "A Variable Order RungeKutta Method for Initial Value Problems with Rapidly Varying Right-hand Sides," *ACM Transactions on Mathematical Software*, vol. 16, pp. 201 222, 1990.

[2] C. Conley, *Isolated invariant Sets and the Morse Index*, CBMS **38**. AMS, Providence, RI. 1978.

[3] *Dynamics and Bifurcations*, Texts in Applied Mathematics **3**, Springer-Verlag, New York 1991.

[4] Helwig Hauser, Robert S. Laramee and Helmut Doleisch, "State-of-the-Art Report 2002 in Flow Visualization," *TR-VRVis-2002-003 , Technical Report, VRVis Research Center*, January 2002, Vienna, Austria.

[5] James H. Hays, and Irfan Essa, "Image and Video Based Painterly Animation," *NPAR 2004: Third International Symposium on Non-Photorealistic Animation and Rendering*, June 7-9, 2004 Annecy, France (to appear).

[6] James L. Helman, and Lamberms Hesselink, "Visualizing Vector Field Topology in Fluid Flows," *IEEE Computer Graphics and Applications*, Vol. 11, No.3, May, pp.36-46, 1991

[7] Aaron Hertzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes," *ACM SIGGRAPH 1998*, pp. 453-460, July 1998.

[8] Aaron Hertzmann, Dennis Zorin, "Illustrating smooth surfaces," *ACM SIGGRAPH 2000*, pp. 517-526, August 2000.

[9] *Differential Equations, Dynamical Systems, and Linear Algebra*, Academic Press, 1974.

[10] Tomasz Kaczynski, Konstantin Mischaikow and Marian Mrozek, *Computational Homology*, Applied Mathematical Sciences, **157** Springer-Verlag, New York, 2003.

[11] R. Kimmel and J.A. Sethian, "Computing Geodesic Paths on Manifolds," *Proceedings of National Academy of Sciences*, Vol. 95, No. 15, pp. 8431-8435, July 1998.

[12] Stephen Mann, and Alyn Rockwood, "Computing Singularities of 3D Vector Fields with Geometric Algebra," *IEEE Visualization '02*, pp. 283-290.

[13] K. Mischaikow, Topological techniques for efficient rigorous computation in dynamics, *Acta Numerica 2002*, (435-478) Cambridge University Press, 2002.

[14] K. Mischaikow, and M. Mrozek, "Conley Index," *Handbook of Dynamic Systems, North-Holland*, Vol 2, pp. 393-460, 2002.

[15] Emil Praun, Adam Finkelstein, Hughes Hoppe, "Lapped textures," *ACM SIGGRAPH 2000*, pp. 465-470, August 2000.

[16] Alyn Rockwood, and Shoeb Bunderwala, "A Toy Vector Field Based on Geometric Algebra," *Proceeding Application of Geometric Algebra in Computer Science and Engineering, (AGACSE2001)*, pp. 179-185, Cambridge, U.K., July 2001.

[17] Jos Stam, "Flows on Surfaces of Arbitrary Topology," *ACM Transactions on Graphics (SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 724-731, July 2003.

[18] H. Theisel and T. Weinkauf "Vector Field Metrics Based on Distance Measures of First Order Critical Points," *V. Skala (editor): Journal of WSCG 10*, pp:121-128, 2002.

[19] H. Theisel, "Designing 2D Vector Fields of Arbitrary Topology," *Computer Graphics Forum (Proceedings Eurographics 2002)*, 21(3), 2002.

[20] X. Tricoche, "Vector and Tensor Field Topology Simplification, Tracking, and Visualization," *Ph.D. Thesis, Universitt Kaiserslautern*, November, 2002.

[21] Greg Turk, "Texture Synthesis on Surfaces," *ACM SIGGRAPH 2001*, pp. 347-354, August 2001.

[22] Jarke J. van Wijk, "Image Based Flow Visualization," *ACM Transactions on Graphics (SIGGRAPH 2002)*, Vol. 21, No. 3, pp. 745-754, July 2002.

[23] Jarke J. van Wijk, "Image Based Flow Visualization for Curved Surfaces," *In: G. Turk, J. van Wijk, R. Moorhead (eds.), Proceedings IEEE Visualization 2003*, pp. 123-130, October 2003.

[24] Li-Yi Wei and Marc Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces," *ACM SIGGRAPH 2001*, pp. 355-360, August 2001.

[25] W. Welch, and A. Witkin, "Free-form shape design using triangulated surfaces," *ACM SIGGRAPH 1994*, pages 247256, 1994.

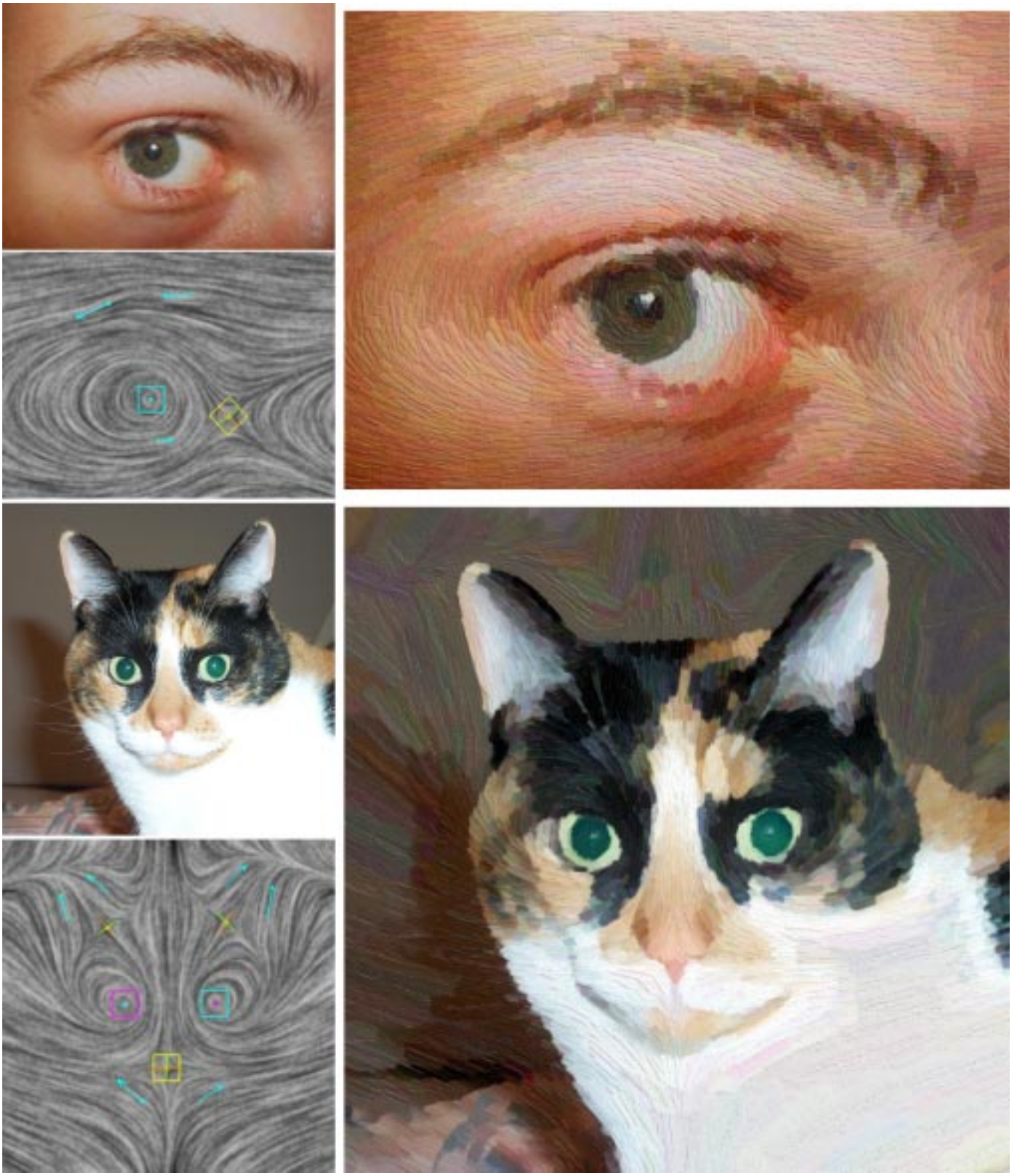

Figure 14: This figure shows the results of applying our planar vector field design system to painterly rendering. The upper and lower portions show the painterly rendering of two input images: an eye and a cat. The input vector fields were created using our system and their visualization are also shown. Note that the high-quality images shown here are produced off-line with the approach of Hays and Essa [5].